**Практическая работа «Перегрузка методов и полифорфизм».**

**Вариант 3**

**Расчет площади квадрата {S=a2}, прямоугольника {S=a\*b}, прямоугольного треугольника {S=a\*b/2}, трапеции {S=(a+b)\*h/2}.**

**Листинг кода с перегрузкой:**

class AreaCalculator

{

public static double CalculateArea(double a)

{

return a \* a;

}

public static double CalculateArea(double a, double b)

{

return a \* b;

}

public static double CalculateArea(double a, double b, bool isTriangle)

{

if (isTriangle)

{

return a \* b / 2;

}

else

{

return a \* b;

}

}

public static double CalculateArea(double a, double b, double h)

{

return (a + b) \* h / 2;

}

static void Main(string[] args)

{

double squareSide = 2;

double squareArea = CalculateArea(squareSide);

Console.WriteLine($"Площадь квадрата со стороной {squareSide} равна {squareArea}.");

double rectangleSideA = 3;

double rectangleSideB = 4;

double rectangleArea = CalculateArea(rectangleSideA, rectangleSideB);

Console.WriteLine($"Площадь прямоугольника со сторонами {rectangleSideA} и {rectangleSideB} равна {rectangleArea}.");

double triangleSideA = 3;

double triangleSideB = 4;

double triangleArea = CalculateArea(triangleSideA, triangleSideB, true);

Console.WriteLine($"Площадь прямоугольного треугольника с катетами {triangleSideA} и {triangleSideB} равна {triangleArea}.");

double trapezoidBaseA = 3;

double trapezoidBaseB = 5;

double trapezoidHeight = 4;

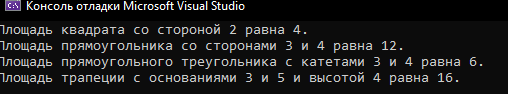
double trapezoidArea = CalculateArea(trapezoidBaseA, trapezoidBaseB, trapezoidHeight);

Console.WriteLine($"Площадь трапеции с основаниями {trapezoidBaseA} и {trapezoidBaseB} и высотой {trapezoidHeight} равна {trapezoidArea}.");

}

}

**Скриншот Результата:**



**Листинг кода с полиморфизмом:**

namespace \_1\_Zadanie\_2

{

abstract class Shape

{

public abstract double CalculateArea();

public virtual void PrintArea()

{

Console.WriteLine($"Площадь фигуры равна {CalculateArea()}.");

}

}

class Square : Shape

{

public double Side { get; set; }

public Square(double side)

{

Side = side;

}

public override double CalculateArea()

{

return Side \* Side;

}

public override void PrintArea()

{

Console.WriteLine($"Площадь квадрата со стороной {Side} равна {CalculateArea()}.");

}

}

class Rectangle : Shape

{

public double Width { get; set; }

public double Height { get; set; }

public Rectangle(double width, double height)

{

Width = width;

Height = height;

}

public override double CalculateArea()

{

return Width \* Height;

}

public override void PrintArea()

{

Console.WriteLine($"Площадь прямоугольника со сторонами {Width} и {Height} равна {CalculateArea()}.");

}

}

class RightTriangle : Shape

{

public double LegA { get; set; }

public double LegB { get; set; }

public RightTriangle(double legA, double legB)

{

LegA = legA;

LegB = legB;

}

public override double CalculateArea()

{

return LegA \* LegB / 2;

}

public override void PrintArea()

{

Console.WriteLine($"Площадь прямоугольного треугольника с катетами {LegA} и {LegB} равна {CalculateArea()}.");

}

}

class Trapezoid : Shape

{

public double BaseA { get; set; }

public double BaseB { get; set; }

public double Height { get; set; }

public Trapezoid(double baseA, double baseB, double height)

{

BaseA = baseA;

BaseB = baseB;

Height = height;

}

public override double CalculateArea()

{

return (BaseA + BaseB) \* Height / 2;

}

public override void PrintArea()

{

Console.WriteLine($"Площадь трапеции с основаниями {BaseA} и {BaseB} и высотой {Height} равна {CalculateArea()}.");

}

}

class Program

{

static void Main(string[] args)

{

Shape[] shapes = new Shape[]

{

new Square(2),

new Rectangle(3, 4),

new RightTriangle(3, 4),

new Trapezoid(3, 5, 4)

};

// Выводим площади всех фигур

foreach (var shape in shapes)

{

shape.PrintArea();

}

}

}

}

**Скриншот Результата:**

